CIAIC-ASC SYSTEM FOR DCASE 2019 CHALLENGE TASK1

Technical Report

Mou Wan, Rui Wang, Bolun Wang, Jisheng Bai, Chen Chen,
Zhonghua Fu, Jianfeng Chen, Xiaolei Zhang, Susanto Rahardja

Northwestern Polytechnical University, Xi' an, China,
{wangmou21, wangrui2018, blwang, baijs, cc_chen524}@mail.nwpu.edu.cn
{mailfzh, chenjf, xiaolei.zhang, susanto}@nwpu.edu.cn

ABSTRACT

In this report, we present our systems for the subtask A and subtask B of the DCASE 2019 Task1, i.e. acoustic scene classification. The subtask A is a basic closed-set classification. Its data was collected from a single device. In our system, we first extracted several acoustic features such as mel-spectrogram, hybrid constant-Q transform, harmonic-percussive source separation, etc. Convolution neural networks (CNN) with average pooling are used to categorize the acoustic scenes. We trained a CNN given each acoustic feature, and integrated the CNNs by averaging their outputs. The subtask B is a classification problem on mismatched devices. For this task, we first introduce a Domain Adaptation Neural Network (DANN) to extract domain-unrelated features, and further aggregated the DANN with the CNN models for better performance. The accuracies of our system for the subtask A are 0.783 on the public validation dataset and 0.816 on the leaderboard dataset respectively. The accuracy for the subtask B is 0.717 on the leaderboard dataset, which shows the effectiveness of our method on the cross-domain problem.

Index Terms— DCASE 2019, acoustic scene classification, domain adaptation neural network, convolution neural network.

1. INTRODUCTION

Sounds carry a large amount of information about our everyday environment and physical events [1]. Developing signal processing methods to extract this information automatically has huge potential in several applications, such as information retrieval [2], mobile devices, robots, etc. Acoustic scene classification (ASC) [3] is one of such ongoing research subjects. Its aim is to classify audio recordings, recorded in a public area, into predefined acoustic scene classes. It is an important task for human [4]. If a computer can automatically recognize acoustic scenes, it can be applied to various fields, such as robotic navigation [5], context-aware services [6], surveillance [7] and etc.

Detection and classification of acoustic scenes and events (DCASE) hosted by IEEE audio and acoustic signal processing (AASP) is a series of challenges aimed at developing sound classification and detection systems, in which the dataset is open and the evaluation system is provided [8]. It is also one of the first large-scale challenges of ASC research. In DCASE 2019, ASC is under the task 1, which includes 3 subtasks viz: a matching device ASC subtask, a mismatching device ASC subtask and an open set ASC subtask.

A number of novel approaches had been proposed in previous DCASE challenges. Most of the submissions in those challenges were based on hand-made features, such as Mel frequency cepstral coefficients (MFCC) [9][10], linear prediction cepstral coefficient [11] and perceptual linear predictive [11]. Some features that are widely used for image processing, such as the histogram of gradients and local binary patterns [12], are also good presentations of acoustic scenes. In DCASE 2017 and DCASE 2018, log-mel energies [4][13][14] and its harmonic-percussive source separation (HPSS) [4][15] were two main features used in most submissions. Beside, constant Q transformation (CQT) were also employed such as in [16][14]. However, we find that most of those works focused on developing powerful classifiers, while little attention was paid on discriminant acoustic features.

In early challenges, conventional machine learning models are used, such as Gaussian mixture model [10] and support vector machine [17], as well as conventional i-vectors [9] and non-negative matrix factorization [18]. Recently, the state-of-the-art methods in the DCASE tasks are based on convolution neural networks (CNN) [4][14][15][19][20]. Specially, Kong et al. [20] proposed generic cross-task baseline systems based on CNN for all tasks of D-CASE 2019. The CNN based methods usually take the log-mel spectrograms of audio recordings as the acoustic features in either the frame-level or the clip-level.

In the DCASE 2019 challenge, we proposed two different AS-C systems for the subtask A and subtask B respectively. For the subtask A, we first extracted several different features that have complementary information with each other. Then, we fed two-dimensional features into two dimensional CNN (2D-CNN) separately, such as log-mel energies and its HPSS, Hybrid constant-Q transform (HCQT), and fed the original waves into a one-dimensional CNN (1D-CNN) directly. Finally, we fused the outputs of different CNN models for prediction. Subtask B is a classification problem with mismatched domains, which is a cross-channel problem. For the subtask, we proposed a Domain Adaptation Neural Network (DANN) [21] framework to project different domains into one common subspace. Here, we regarded the data from device A as the source domain and the data from devices B and C as the target domains. To extract more information of the acoustic scenes, we chose 64 log-mel energies of the spectrograms as the main feature. Besides, we also adopted CNN for the classification problem of the subtask B, and aggregated DANN and different CNNs that take different input features for prediction.

This technical report is organized as follows. Section 2 presents the framework of our system. Then, Sections 3 and 4 illustrate the features and CNN models in detail. Section 5 presents the experiments. Section 6 concludes this report.

2. FEATURE EXTRACTION

2.1. Log-mel Energies

The log-mel energies is the most popular feature for acoustic scene classification, as well as other tasks in DCASE. It is considered to be the most suitable for acoustic scene classification [4]. Short-time Fourier transform (STFT) can calculate the spectrogram by multiplying the window function frame by frame to look at the time change. First, we extract the spectrogram with STFT. The window function of STFT is a Hann window, the window size is 2048 (64 ms) and hop size is 500 (15ms). Therefore, there are 64 frames in one second. Then, we apply Mel-filter banks on spectrogram to get Mel-energies, where filter bins equal 256 in subtask A and 64 or 128 in subtask B. In addition, cut-off frequencies of Mel-filter are from 50 Hz to 14 kHz. A logarithm operation is further applied to obtain the log-mel energies. Therefore, we can obtain a feature of [640, 256, 1] from one audio recording.

2.2. Harmonic-percussive source separation

Sound can be generally be divided into two types: harmonic and percussive. Han et al. separated the audio clips into two using H-PSS in DCASE 2017 for the first time, which enables to separately exploit harmonic and percussive aspects of a sound [15]. Similarly, we applied HPSS on Mel-spectrogram with the parameters described in subsection 2.1. For HPSS, we used the corresponding function in librosa which is a Python package for music and audio analysis. The initial values in librosa are used for parameters of HPSS algorithm. Then we can obtain data of [640, 256, 2] shape from one audio recording.

2.3. Hybrid constant-Q transform

Constant-Q transform essentially transforms a series of data to the frequency domain. It is similar to the STFT and very closely related to the complex Morlet wavelet transform. CQT is a common feature in music signal processing and acoustic scene analysis. HCQT uses the pseudo CQT for higher frequencies where the hop length is longer than half the filter length. For lower frequencies, HCQT uses the full CQT. It is a more computationally efficient version of CQT. We extract 84-dimensional HCQT with the initial parameter in librosa. Therefore, we get a feature of [640, 84, 1] shape from each audio recording.

3. MODELS

In our system, the classifier is based on CNN. For subtask A, we built two CNN models for acoustic scene classification, i.e. two-dimensional CNN for two dimensional feature and one-dimensional CNN for wave. For subtask B, we built a multi-task neural network with adversarial training. Besides, we built other six CNN models to make a combined prediction.

3.1. Two-dimensional CNN

We followed the CNN framework proposed in [20], as is illustrated in Tab. 1. Inspired from VGG network, we built a convolution block using two cascaded convolution layers with 3×3 kernels. Each convolutional operation was followed by batch normalization to speed up and stabilise the model training. After batch normalisation, we applied ReLU as non-linearity activation function. For each convolutional block, we used average pooling with a size of 2×2 in subtask A. We also used max pooling in subtask B. Finally, a fully connected layer with softmax nonlinearity was applied to predict acoustic scene for each recording. Because ASC is a classification task, we chose cross entropy loss to train the network.

Table 1: Two-dimensional CNN

Input: features
(3x3) Conv2D-64-BN-ReLU
(3x3) Conv2D-64-BN-ReLU
(2x2) Average Pooling
(3x3) Conv2D-64-BN-ReLU
(3x3) Conv2D-64-BN-ReLU
(2x2) Average Pooling
(3x3) Conv2D-64-BN-ReLU
(3x3) Conv2D-64-BN-ReLU
(2x2) Average Pooling
(3x3) Conv2D-64-BN-ReLU
(3x3) Conv2D-64-BN-ReLU
(2x2) Average Pooling
Dense-10-SoftMax

3.2. One-dimensional CNN

Considering the aforementioned features are extracted based on frequency domain, the wave is a signal of time domain, which will offer complementary information. Therefore, we built a 1D-CNN and use raw audio directly to train it. First, we resampled all audio samples at 16 kHz. The CNNs classified the raw audio using one-dimensional convolution along the time. Tab. 2 shows the network architecture. Similar to 2D-CNN, we also built a convolution block using two cascaded convolution layers. Batch normalization and ReLU function were used after each convolutional operation. In the first convolution block, we used big kernel size and pooling size because the signal in time domain has much redundant information. After four convolution blocks, we took a dropout with 0.2 rate to avoid overfitting. Finally, two dense layers were applied to predict acoustic scene for each recording.

3.3. Domain Adaptation Neural Network

Subtask B is a cross-channel problem. We proposed to project data from the different domains into one common subspace to mitigate domain mismatch. The framework of our system for subtask B is illustrated in Fig. 1. It comprises three parts, including feature extractor, scene predictor and domain predictor. This is realized by training a multi-task neural network that learns a scene-discriminative and domain-invariant feature representation, which DANN mainly does.

DANN is different from traditional feed-forward neural network which has single input and single output, but is similar to the multi-task neural network. In the structure of DANN, there is one input layer and two output layers. We assume that the input data as $\mathbf{x} \in$ X, scene label $\mathbf{y} \in$ Y and domain label $\mathbf{d} \in \{[0,1],[1,0]\}$, where X and Y are input space and output space. We assume that there are two different domains, source domain S and target domain T. Denote with $\mathbf{d}_i$ for the *i*-th sample, indicating which domain the

Table 2: One-dimensional CNN

Input: wave
(9x1) Conv1D-16-BN-ReLU (9x1) Conv1D-16-BN-ReLU (16x1) Average Pooling
(3x1) Conv1D-32-BN-ReLU (3x1) Conv1D-32-BN-ReLU (10x1) Average Pooling
(3x1) Conv1D-32-BN-ReLU (3x1) Conv1D-32-BN-ReLU (10x1) Average Pooling
(3x1) Conv1D-256-BN-ReLU (3x1) Conv1D-256-BN-ReLU Global Average Pooling
Dropout (0.2)
Dense-64-ReLU
Dense-10-SoftMax

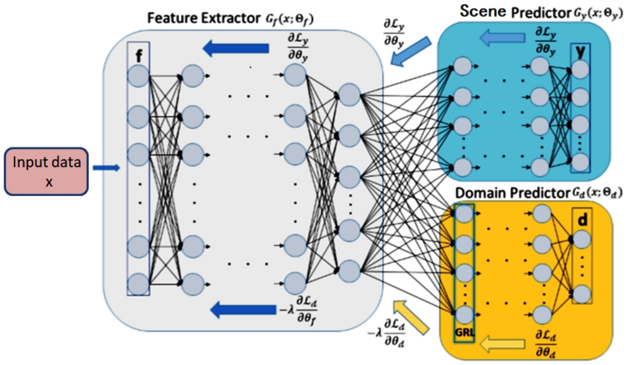
Feature Extractor $G_f(x;\theta_f)$
Scene Predictor $G_y(x;\theta_y)$
Domain Predictor $G_d(x;\theta_d)$
Input data x
f
y
d
GRL
$\frac{\partial \mathcal{L}_y}{\partial \theta_y}$
$-\lambda\frac{\partial \mathcal{L}_d}{\partial \theta_f}$
$-\lambda\frac{\partial \mathcal{L}_d}{\partial \theta_d}$
$\frac{\partial \mathcal{L}_d}{\partial \theta_d}$

Figure 1: Framework of DANN.

data comes from. If $\mathbf{x}_i$ is from source domain, its domain label will be $[1, 0]$, otherwise its domain label will be $[0, 1]$.

The feature extractor aims to find one representation which is discriminative in different scenes while indiscriminating in different domains. The scene classifier and domain classifier mainly help train the feature extractor, but the former also plays an important role in predicting the scene label in ASC. We assume that the feature extractor as G_f with parameters θ_f, the scene classifier as G_y with parameters θ_y, the domain classifier as G_d with parameters θ_d. In the training process of DANN, the three parts are trained simultaneously and we want to find suitable θ_f to minimize the scene classification loss while maximize the domain classification loss. This can be achieved with the help of a gradient reversal layer which is between the feature extractor and the domain classifier and can find the saddle point between scene classifier and domain classifier. Generally, we need to search a positive hyper-parameter to multiply the gradient reversal layer to make a balance between losses of the two classifiers. For two classifiers, we need to search θ_y and θ_d to minimize the prediction losses.

We define the loss function of DANN for source data as:

$$\begin{aligned} E(\theta_f, \theta_y, \theta_d) &= \sum_{i=1}^{N} L_y(G_y(G_f(x_i;\theta_f);\theta_y), y_i) - \\ &\quad \lambda \sum_{i=1}^{N} L_d(G_d(G_f(x_i;\theta_f);\theta_d), d_i) \qquad (1) \\ &= \sum_{i=1}^{N} L_y^i(\theta_f, \theta_y) - \lambda \sum_{i=1}^{N} L_d^i(\theta_f, \theta_d) \end{aligned}$$

where $d_i = [1, 0]$, L_y^i is the loss of the *i*-th training sample for scene label and L_d^i is the loss of *i*-th training sample for domain label. We select different loss functions for them: cross entropy for the former and Mean Square Error (MSE) for the latter.

Based on our idea, we are seeking a set of parameters $\hat{\theta}_f$, $\hat{\theta}_y$, $\hat{\theta}_d$ that deliver a saddle point of the function Eq. 1:

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d) \qquad (2)$$

$$\hat{\theta}_d = \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d). \qquad (3)$$

θ_y can be optimized by minimizing Eq. 1, while θ_d can be optimized by maximizing the equation. We optimize θ_f by minimizing the first item while maximizing the second item. As for updates of the parameters, we choose adaptive moment estimation (Adam) approach. In this way, we can make sure that the extracted feature are scene-discriminative and domain-invariant.

3.4. Fusion

To obtain better performance, we adopted an ensemble strategy, i.e. fusing the output of different methods, including different features and CNN with different topologies. For subtask A, we used all of aforementioned features. We fed four features into 2D-CNN, i.e. log-mel spectrogram, HPSS, HCQT and MFCC, and fed the raw wave into 1D-CNN directly. Finally, we average the outputs of those methods in probability space.

For subtask B, we fed two features into CNN, i.e. log-mel spectrogram and HCQT. Then we fused the outputs of CNN with different settings, where the layers ranged from 5 to 13 with interval of 2 and pooling layer can be average pooling or max pooling.

4. EXPERIMENTS

4.1. Dataset

The dataset for ASC task is the TAU Urban Acoustic Scenes 2019 dataset, consisting of recordings from various acoustic scenes [1]. The recordings are recorded from different locations in 12 European cities with different devices. There are 10 acoustic scenes, including airport, shopping mall, metro station, pedestrian street, public square, street with traffic, tram, bus, metro and urban park. The audio dataset includes three different versions: TAU Urban Acoustic Scenes 2019, TAU Urban Acoustic Scenes 2019 Mobile and TAU Urban Acoustic Scenes 2019 Openset. The first dataset is used for subtask A where the development and evaluation are recorded with device A. It contains 40 hours of data. The length of each recording is 10s, therefore we have 14400 segments. In test set, there are 7200 segments to predict.

The second dataset is used for subtask B where the material is recorded with devices A, B and C. Data from device A is resampled and averaged into a single channel, to align with the properties of the data recorded with devices B and C. For subtask B, the development data set contains in total 46 hours of data, with 16560 segments, of which 14400 from device A, 1080 from device B and 1080 from device C. In the training set, there are 10625 segments in all with 9185 for device A, 540 for device B and 540 for device C. In the test set, there are 5265 segments in all with 4185 for for device A, 540 for device B and 540 for device C.

The DCASE 2018 dataset is recorded using binaural microphone. We first got mono audio by averaging the two channels. Then, the DCASE 2019 dataset was sampled at 48 kHz. Learned from [20], 32 kHz sampling rate can contain most energy. For both subtask A and subtask B, we resampled all audio recordings at 32 kHz.

4.2. Baseline System

The official baseline system implements a convolutional neural network (CNN) based approach. Log-mel energies are first extracted from each recording, and then a network consisting of two CNN layers and one fully connected layer is trained to assign scene labels to the audio signals. The detailed structure is shown in [1]. For both subtask A and subtask B, the evaluation criterion is classification accuracy (ACC), which is obtained by averaging the class-wise accuracy of all sound classes.

4.3. Result

The table below shows the development or leardboard performance of different models. The official baseline system result is from official website of DCASE [1] and leaderboard in Kaggle [22][23].

The result of subtask A is shown in Tab. 3. In subtask A, the CNN with log-mel energies achieves an accuracy of 0.721, outperforming the system in [20] and official baseline system. Our fusion system achieves an accuracy of 0.835 and tied for fourth place.

In subtask B, we combined seven models, including DANN and other six CNN models to make predictions on the three features respectively. We made a fusion on the predictions, which achieved 0.717 on leaderboard, as shown in Tab. 4.

Table 3: Comparison results of subtask A between different methods and feature types.

Methods	ACC on validate	ACC on leaderboard
Baseline system	0.625	0.643
Mel-2D-CNN	0.721	
HPSS-2D-CNN	0.724	
HCQT-2D-CNN	0.679	
MFCC-2D-CNN	0.663	
wave-1D-CNN	0.570	
Fusion	0.795	0.835

Table 4: Comparison results of subtask B between different methods and feature types.

Methods	ACC on leaderboard
Baseline system	0.480
64 log-mel-combined	0.675
128 log-mel-combined	0.707
HCQT-combined	0.667
Fusion	0.717

5. CONCLUSION

In this paper, we described our systems for subtask A and subtask B of DCASE 2019 task1. For subtask A, we extracted a set of features and fed them into CNN separately. For subtask B, we fed log-mel energies and HCQT into two CNN models, i.e. 2D-CNN and DANN. Finally, we made fusion on outputs of different methods. Experimental results showed our systems outperformed baseline systems.

6. REFERENCES

[1] http://dcase.community/challenge2019/.

[2] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Semantic annotation and retrieval of music and sound effects,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, Feb 2008.

[3] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13.

[4] Y. Sakashita and M. Aono, “Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions,” DCASE2018 Challenge, Tech. Rep., September 2018.

[5] S. Chu, S. Narayanan, C. . J. Kuo, and M. J. Mataric, “Where am i? scene recognition for mobile robots using audio features,” in *2006 IEEE International Conference on Multimedia and Expo*, July 2006, pp. 885–888.

[6] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, “Audio-based context recognition,” *IEEE Transactions on Audio,*

Speech, and Language Processing, vol. 14, no. 1, pp. 321–329, Jan 2006.

[7] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, Oct 2005, pp. 158–161.

[8] T. Heittola and A. Mesaros, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," DCASE2017 Challenge, Tech. Rep., September 2017.

[9] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[10] G. Takahashi, T. Yamada, S. Makino, and N. Ono, "Acoustic scene classification using deep neural network and frame-concatenated acoustic feature," DCASE2016 Challenge, Tech. Rep., September 2016.

[11] S. Park, S. Mun, Y. Lee, and H. Ko, "Score fusion of classification systems for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[12] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of timecfrequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 142–153, Jan 2015.

[13] S. Mun, S. Park, D. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane," DCASE2017 Challenge, Tech. Rep., September 2017.

[14] H. Zeinali, L. Burget, and H. Cernocky, "Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge," DCASE2018 Challenge, Tech. Rep., September 2018.

[15] Y. Han and J. Park, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," DCASE2017 Challenge, Tech. Rep., September 2017.

[16] Z. Weiping, Y. Jiantao, X. Xiaotao, L. Xiangtao, and P. Shaohu, "Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion," DCASE2017 Challenge, Tech. Rep., September 2017.

[17] B. Elizalde, A. Kumar, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, "Experiments on the DCASE challenge 2016: Acoustic scene classification and sound event detection in real life recording," DCASE2016 Challenge, Tech. Rep., September 2016.

[18] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Supervised nonnegative matrix factorization for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[19] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and I-vectors," DCASE2018 Challenge, Tech. Rep., September 2018.

[20] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.

[21] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4889–4893.

[22] https://www.kaggle.com/c/dcase2019-task1a-leaderboard/.

[23] https://www.kaggle.com/c/dcase2019-task1b-leaderboard/.